New Human-Computer Interface Concepts for Mission Operations

Jeffrey A. Fox, Mary Sue Hoxie, Dave Gillen, Christopher Parkinson
Pacific Northwest National Laboratory*
901 D Street, SW, Suite 900
Washington, DC 20024
jeff.fox@pnl.gov, mary.sue.hoxie@pnl.gov, david.gillen@pnl.gov, christopher.parkinson@pnl.gov

Julie Breed, Stephanie Nickens
NASA/Goddard Space Flight Center
Greenbelt Road
Greenbelt, Maryland 20771
julie.breed@gsfc.nasa.gov, Stephanie.H.Nickens.1@gsfc.nasa.gov

Mick Baitinger
8403 Colesville Road, Suite 915
Silver Spring, Maryland 20910
NEXGEN Solutions, Inc.
mbating@ngsinc.com

The current climate of budget cuts has forced the space mission operations community to reconsider how it does business. Gone are the days of building one-of-kind control centers with teams of controllers working in shifts 24 hours per day, 7 days per week. Increasingly, automation is used to significantly reduce staffing needs. In some cases, missions are moving towards lights-out operations where the ground system is run semi-autonomously. On-call operators are brought in only to resolve anomalies. Some operations concepts also call for smaller operations teams to manage an entire family of spacecraft. In the not too distant future, a skeleton crew of full-time general knowledge operators will oversee the operations of large constellations of small spacecraft, while geographically distributed specialists will be assigned to emergency response teams based on their expertise.

As the operations paradigms change, so too must the tools to support the mission operations team's tasks. Tools need to be built not only to automate routine tasks, but also to communicate varying types of information to the part-time, generalist, or on-call operators and specialists more effectively. Thus, the proper design of a system's user-system interface (USI) becomes even more important than before. Also, because the users will be accessing these systems from various locations (e.g., control center, home, on the road) via different devices with varying display capabilities (e.g., workstations, home PCs, PDAs, pagers) over connections with various bandwidths (e.g., dial-up 56k, wireless 9.6k), the same software must have different USIs to support the different types of users, their equipment, and their environments. In other words, the software must now adapt to the needs of the users!

This paper will focus on the needs and the challenges of designing the Spacecraft Emergency Response System (SERS), a Web-centric suite of tools built to enable lights-out operations. More specifically, SERS is a collaborative environment that enables secure distributed fault management though the effective use USIs that are optimized for a wide variety of devices, including traditional PC Web browsers, pagers, telephony, and Internet phones. SERS has become a widely adopted tool for operations, and its success is directly related to its utility and the usability (ease of use) of its USIs.

Importance of the User-System Interface (USI)

A key factor in the success of any software product is its USI, because the end-user views all aspects of the system through the interface. The effectiveness of a USI design generally can be expressed in terms of: (1) time to learn, (2) speed of performance, (3) rate of user errors, (4) retention over time, and (5) subjective satisfaction [1]. The more complex the system, the greater the need for an effective user interface [2, 3]. Some of the more important factors that should influence the USI design are

*Pacific Northwest National Laboratory is operated for the U.S. Department of Energy by Battelle under Contract DE-AC06-76RLO 1830.

shown in Figure 1. The factors can be grouped into three larger components: the user, the implementation of the system, and the task and environment. These factors are interrelated and the level of impact of any individual factor depends on the specifics of the system.

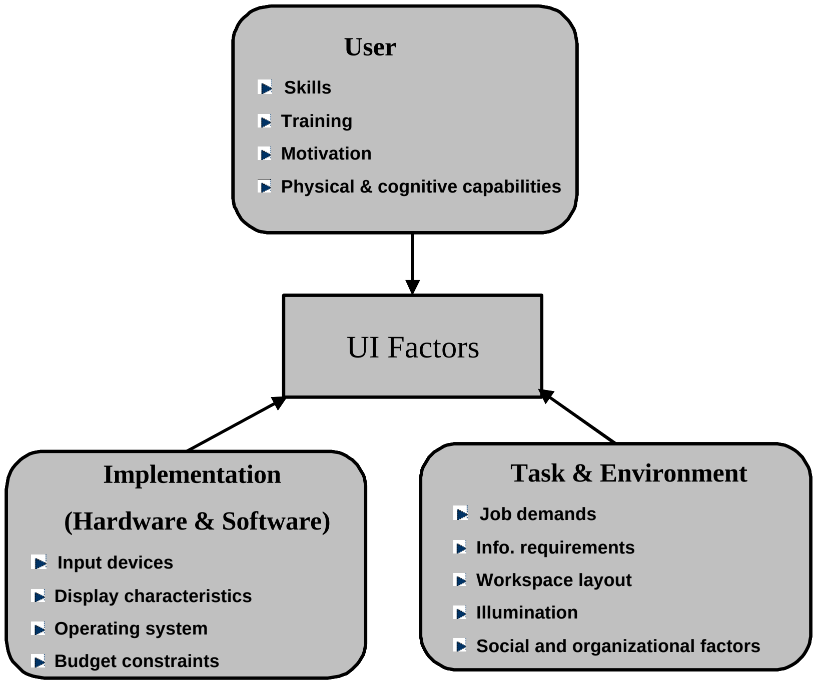

Figure 1. Factors Influencing a System's USI

NASA Goddard's Advanced Architectures and Automation (AAA) branch understands the importance of the user interface for the success of a complex system, and thus encourages a user-centered design (UCD) approach to system's development. The goal of UCD is to ensure that users' capabilities and limitations are considered in systems design, by involving users in all phases of the design processes.

UCD has guided all phases of the development of the SERS. With SERS, continuous human monitoring is no longer needed during routine operations. Instead, SERS enables "lights-out operations," in which standard process monitoring and management tasks are performed autonomously using advanced automation, expert systems, and software agents. However, when SERS identifies a potential fault or an emergency requiring human attention, a suite of workgroup computing tools dynamically creates a response team by contacting the most appropriate personnel and facilitating their interactions. Thus, SERS implements a proactive management-by-exception paradigm, which better utilizes expensive human and computer resources.

More specifically, SERS is a Web-based collaborative environment that enables secure distributed fault management. SERS is built primarily in Lotus Domino™ and incorporates the use of intelligent agents, threaded discussions, workflow, database connectivity, and links (gateways) to a variety of communications media (e.g., 2-way paging, PDA's, telephony) via commercial gateways. When SERS detects a problem, it automatically:

- Contacts appropriate on-call personnel,
- Synthesizes and presents status and history of events and actions for rapid resolution,
- Automatically generates all necessary reports and documentation, and
- Enables cooperative work by on-call personnel at distributed locations.

More information about SERS, including its development team members, papers and presentations can be found at http://aaaprod.gsfc.nasa.gov/vmoc/SERS/sers.html.

SERS Unique User Interface Challenges

Because SERS' primary goal is to enable lights-out mission operations, the USI for SERS had to meet the following criteria:

1. Easily accessible – The on-call team members could be almost anywhere (in the control center, in an office, at home, or on the road) so they need software that can be used on a variety of devices and at many locations.
2. Easy to use – Because of reduced staffing, the usability of the software is paramount to minimize the workload of the users.
3. Functional – The software cannot trade core capabilities to meet the above two requirements.

Although the main goal of lights-out operations is to reduce costs, it does not provide an excuse to lower efficiency or effectiveness. As the space community well knows, a faster and cheaper approach to operations does not reduce public criticism for mission failure (see Mars explorer [4]). Thus, the demands placed upon software to support lights-out operations is greater than that of traditional operations software. For example, an emergency response system must accomplish traditional operational tasks in addition to unique new capabilities, such as:

1. Provide context – In traditional 7x24 operations, humans are continuously monitoring the health and safety of the spacecraft. In this environment, the sharing of pertinent information is critical to mission success. Team members on the same shift discuss problems, goals, and issues. During shift transitions, teams debrief, noting open issues and items that need to be watched. In lights-out operations, when on-call team members are alerted, they must rapidly come up to speed on the problem to determine the impacts of the information conveyed in the alert message. SERS' USI must quickly convey all the necessary contextual information so that the team member can make an accurate assessment quickly.
2. Support multiple devices – Currently, lights-out teams carry a wide variety of wireless devices (e.g., pagers, cell phones, Internet phones) for alert notifications. SERS must account and compensate for the capabilities and limitations of such devices (e.g., screen size, transmission bandwidth) to properly format the data in an equally effective manner across platforms.
3. Support multiple goals – A single SERS system must support not only the needs of operators checking on the status of a spacecraft, but also the engineers doing testing, diagnostics, and fault recovery, and the mission managers who need to make overall assessments of a mission and its associated personnel.
4. Reduce training – An early realization for the SERS design staff was the need to create software that was not only easy to use, but also easy to learn. The reduced operations team did not have the time or the money to invest in extensive training.
5. Support multiple spacecraft – To reach its cost savings objectives, NASA Goddard needed the ability to have a single operations crew support multiple missions. Thus, SERS needed to be able to integrate and display health and safety data for an entire family of spacecraft (e.g., all small explorer missions, SMEX, missions).

To meet these challenges, SERS was designed, built, and tested using a user-centered design framework. Operations staff were involved in the SERS project from early concept definition through post-deployment usability testing. A variety of design techniques were used in an iterative design cycle, including: (1) cooperative prototyping, (2) focus groups, (3) scenario-based design, (4) usability testing, and (5) task analyses [5, 6]. The next sections of this paper describe challenges and design solutions for three of SERS' core capabilities: (1) basic monitoring and reporting, (2) tracking families of spacecraft, and (3) alert notification and response.

Basic Monitoring and Reporting via the World-Wide Web

After much deliberating, prototyping, and gathering input from steering committee meetings, the SERS design team chose to implement the core SERS USI as a World-Wide Web application, making it accessible from any modern browser. Only a Web browser provides very nearly ubiquitous access. As long as people have access to the Internet, they can access SERS.

Creating SERS as a Web application not only satisfied the SERS' requirement for wide access, but also partially addressed the usability issue. By the time SERS was fielded in April 1998, all of its users had extensive experience using the Web. This significantly reduced training time, compared to learning the interface to a new application. However, using the Web does not guaranty a usable application. There are many unusable sites that are cluttered, poorly structured, or present the wrong information. Developing an intuitive Web application still requires the application of UCD principals.

For SERS, the Web interface needed to hide the underlying complexity of the application (15 databases, multiple communications gateways, multiple commercial applications). The development team created a single simple and consistent USI that used common office metaphors (e.g., tabs across the top of the screen to represent each database). SERS uses the same "look and feel" throughout the application. For example, the main Web page for SERS' most commonly accessed database (Events) is shown in Figure 2. The common elements of that interface are described below.

- The *Option Tabs* allow users to navigate among databases within a category.
- The *Data View* is located on the right two thirds of the screen. The View generally comprises the largest area on the screen, and provides space for the actual data to be displayed. Databases can have multiple Views, each sorting the data in a different manner (e.g., view anomaly reports by date or by spacecraft).
- The *Navigator* is located at the far left of the page. It consists two rows of navigational icons; the first row for access to each of SERS' functional areas – operations, configuration, tools, and pre-mission test and the second row for functions available to all databases – home, print, find, and help. Under the icons is a menu of the *Views* available for that database.
- *Action Bars* are located both above and below the *View* area. They provide links to perform actions on the displayed *View*.

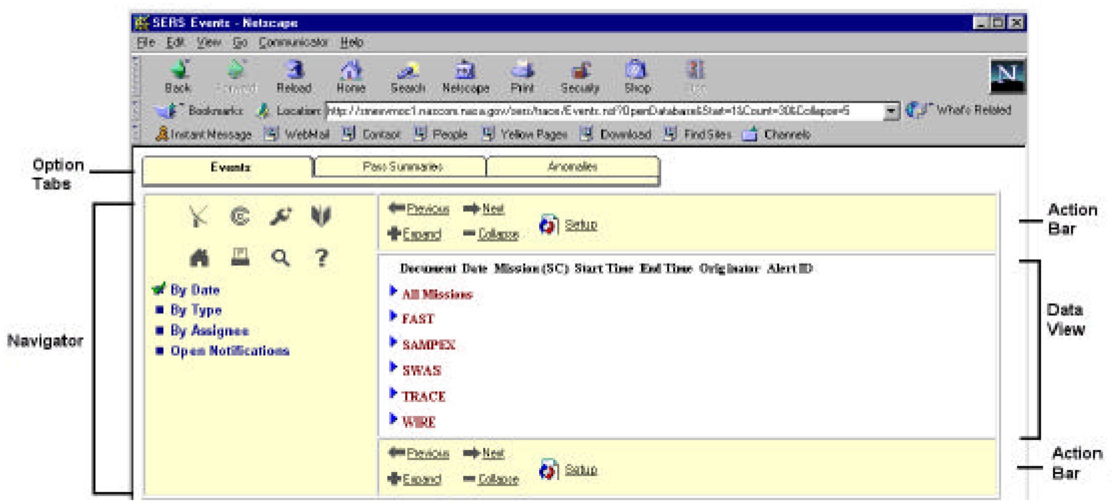

Figure 2. Common SERS USI Database Design

Another technique SERS uses to reduce complexity and information overload is known as "progressive disclosure." Towards that goal, SERS provides greater and greater levels of details as the user needs them, based on the actions and goals. This prevents overloading a user by presenting all of the available data at once.

In SERS, information is disclosed via the use of expanding categories of data within the *Views*. The concept is shown in Figures 2 and 3. Figure 2 shows the fully collapsed list of all the spacecraft being monitored by the SERS SMEX machines. In Figure 3, the user expanded first the TRACE mission (by clicking on the arrow to the left of its name) to display the TRACE event reports. Next the user expanded Pass Event Report #73511 to see who was notified about that event and who responded to the notifications. To see the detailed information contained in the report (e.g., out-of-range mnemonics), the user clicks directly on the report name (see [7] for details of SERS reporting).

Figure 3. Expanded View of TRACE Event Report #73511

Tracking Families of Spacecraft
While SERS' primary Web interface has proven effective for monitoring several spacecraft (based on interviews with operations staff and an independent usability study), the progressive disclosure technique can be slow and cumbersome for a user who wants to ascertain quickly the status of a whole family of spacecraft and the associated ground system. This is a challenge created by the near complete lights-out operations concept being used for the SMEX and MIDEX families of spacecraft, (one-shift: Monday through Friday, 8 hours per day).

At the beginning of a shift an operator needs to assess the overall state of operations quickly without having to click though many hierarchies of menus. To aid in this function, the SERS team developed a new visualization tool called the Focus Area for SERS Tracked Resources (FASTR). FASTR is a highly-tairlorable Java applet, and an HTML alternative for low-bandwidth connections. A Java version of FASTR is shown in Figure 4. FASTR presents high-level status information on each spacecraft being monitored as well as the SERS' servers (not shown). FASTR can display the status of passes (historical and predictive), events, alerts, and responses to the alerts in color-coded cells (informational – blue, OK – green, warning – yellow, and problem – red) in a matrix (right side of screen shown in Figure 4) to aid quick visual scanning. To see the detailed SERS information associated with any of these items, a user simply clicks on the content of the cell, and FASTR displays a new window that contains the associated SERS report

Unfortunately, even such a concise representation of the data can still produced a very long matrix of information that must be scrolled through to see the status of every resource being monitored. As a result, the user many not notice a problem for a spacecraft because its data is off the screen. To compensate for this, an overview visualization is displayed in the left side of the screen (Figure 4). In this area, the entire matrix of information is displayed, but in a reduced scale, so that all the information is always visible. The detailed information is removed, leaving just the color-coded status information. Additionally, the red square shown in that figure acts as a scroll control mechanism for the main viewing area (similar to the "thumb" of a standard scroll bar). As the user moves the square to an area of interest (e.g., a line that shows red), the main display will show that same area, but with its more detailed information. Since different users may want to view different sets of data, FASTR provides preferences for the user to select: (1) which resources to monitor, (2) which parameters to monitor, (3) update frequency, and (4) color intensities, Figure 5.

Alert Notification and Response via Mobile Devices
Though SERS' monitoring and reporting capabilities greatly enhance lights-out operations, it is SERS' emergency alert notification and response functionality that enables this paradigm. Upon detecting a problem with a spacecraft, SERS determines whom to notify based on the type of problem and the

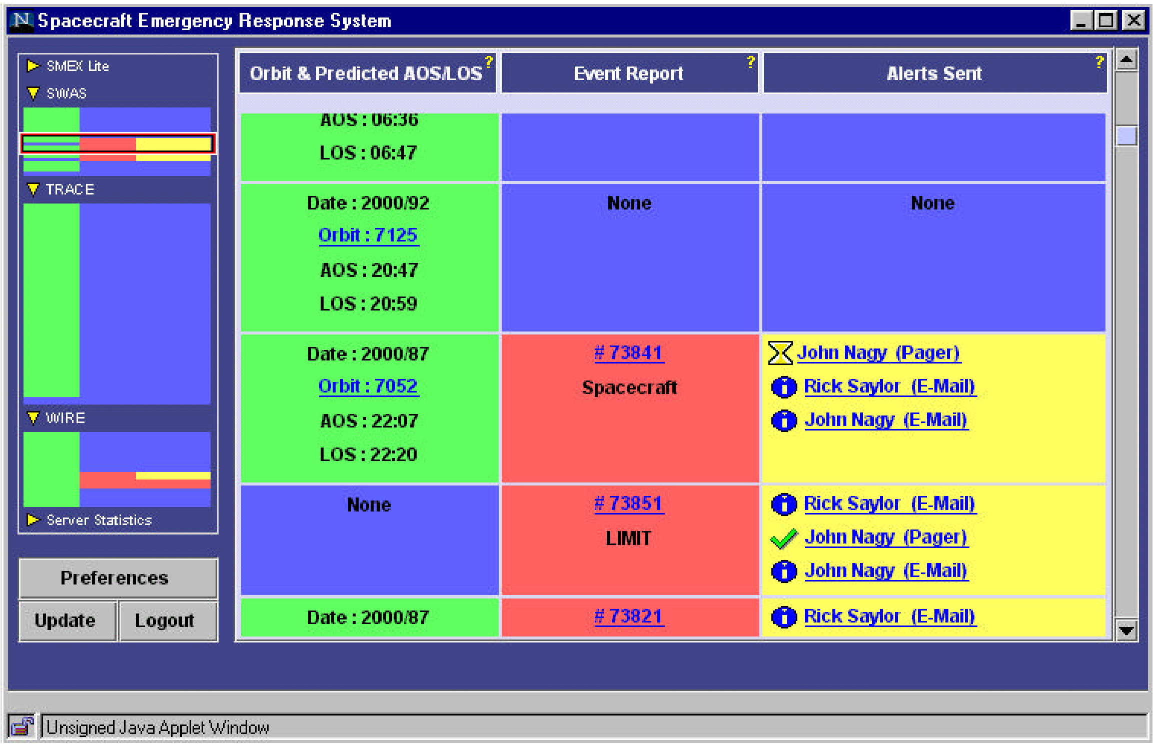

Figure 4. FASTR Applet (Java Version)

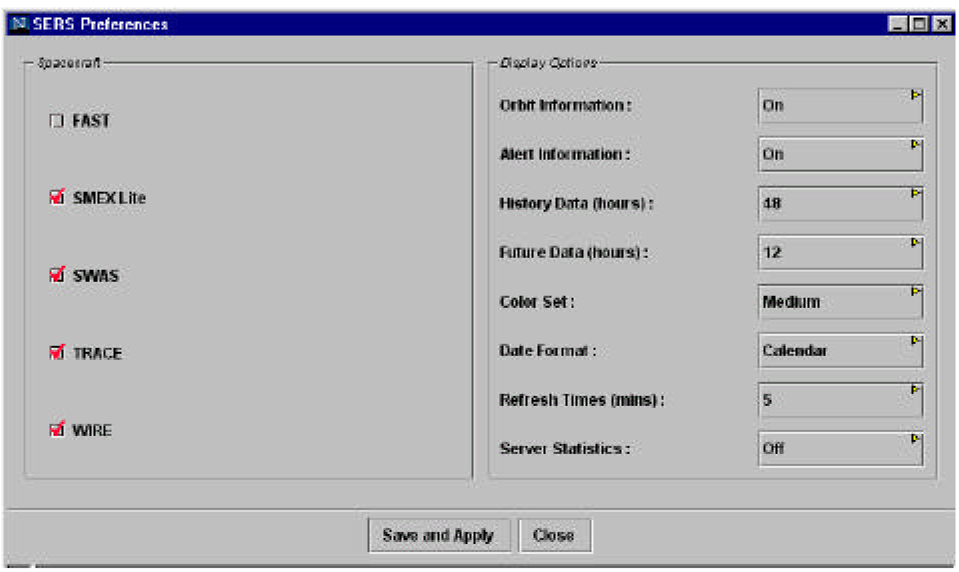

Figure 5. FASTR Customization Screen

skills and availability of the on-call team members. How SERS contacts the team members depends on the severity of the problem and the team member's preferred device: 2-way pagers, telephone, or Internet phone. Each of these devices complies with the SERS requirement that it is 2-way; it can both receive an alert and transmit data/commands back to SERS to trigger additional workflow activities (such as contacting a backup team member or re-scheduling a pass). However, each device introduces its own unique design challenges and user interface design solutions.

Two-Way Pager

For the SMEX missions at Goddard, the primary notification/response devise is the AccessLink™ 2-way pager over the SkyTel™ 2-way wireless network. This combination of device and network allows

SERS to send relevant data to the pager along with “canned” responses (e.g. “accept responsibility” or “defer to backup”). A picture of the pager is shown in Figure 6. A typical SERS alert notification and response sequence is shown Figures 7 through 12. The following are the most challenging design constraints for this paging device, along with the implemented USI solutions(s).

Challenge: Pager’s small 4-line, 20-character scrolling display.
Solution: User can select (via check boxes on SERS website) which elements of the event report (e.g., limit violations, details, etc.) are sent to the pager for each type of alert. This allows the user to tailor the message based upon his/her individual needs.

Challenge: Software keyboard makes it difficult to type in reply (response messages)**,** Figure 13.
Solution: Pre-programmed response options are embedded at the end of the alert message. Via this mechanism, the user scrolls to the appropriate response and selects that option. The pre-programmed responses are specific to the alert notification being sent, such as those shown in Figure 12.

Figure 6. AccessLink Pager with SkyTel Service

Figure 10. SERS Alert Message – Part 4

Figure 7. SERS Alert Message – Part 1

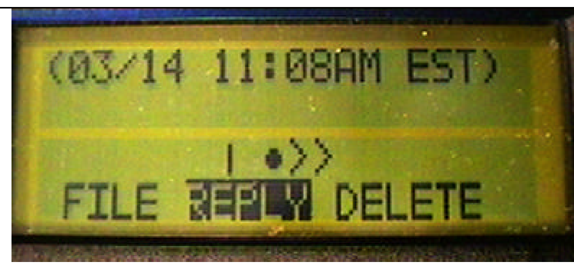

Figure 11. SERS Alert Message – Reply

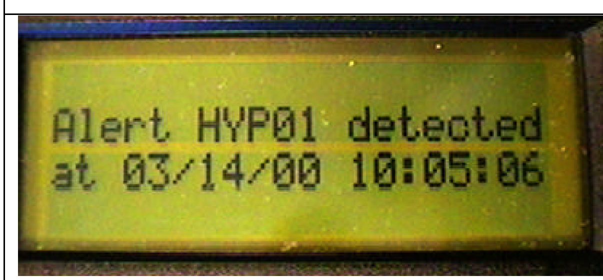

Figure 8. SERS Alert Message – Part 2

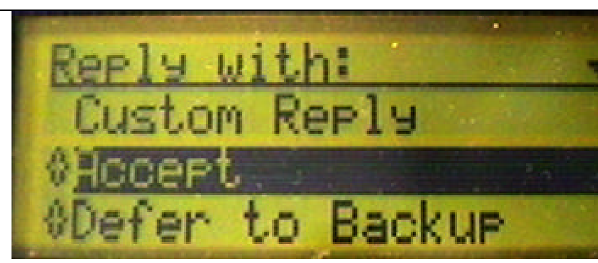

Figure 12. SERS Alert Message – Reply Options

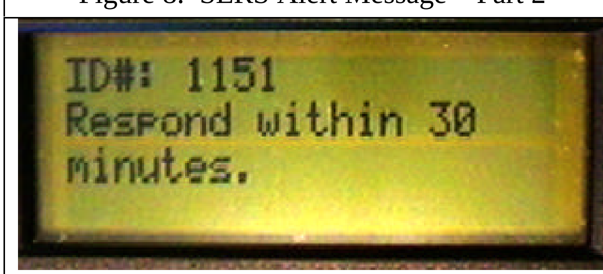

Figure 9. SERS Alert Message – Part 3

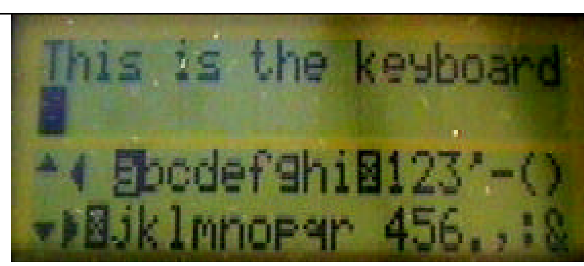

Figure 13. AccessLink Pager Soft Keyboard

Challenge: SkyTel can only transmit 500 characters per page.

Solution: If the alert notification contains more than 500 characters, SERS automatically sends only the first 500 characters, plus a pre-programmed option to receive "More" data. This option will send a message to SERS to retrieve and send out the next block of data. Because of the limited bandwidth of wireless data transmissions (9.6k – 19.2k), it is often better to send the minimum required data.

Although the pager prevents the formatting of the messages from being optimal, the results of this technique have been quite good. Through interviews, multiple operators have stated that they can make first-cut decisions based on the pager information. When additional information is necessary, they then view the detailed reports via a web browser on their PCs.

Telephony

Despite the success of the pager scheme of the SMEX missions, other missions plan to rely more heavily on SERS's telephony USI. The telephony interface is similar to the telephone systems used by credit card companies, mail order firms, and many office messaging system. The basic interaction technique is to use pre-recorded (digitized) messages providing menu options with corresponding telephone input keys. The user presses the appropriate key to select the desired menu option (e.g., press 1 to select the MAP spacecraft). Dynamic data are looked up in the appropriate database and then run through a text-to-speech engine so that they can be "read" over the phone.

The advantages of using telephony over a pager are: (1) that the team member does not have to worry about being in "coverage" of the pager service, since SERS can call regular twisted-pair phones, as well as cell phones; and (2) missions need to supply their team members a single device, as opposed to a pager and cell phone. But like the pager, there are challenges to designing the telephony USI. These, along with the SERS solutions, are listed below:

Challenge: Standard telephony systems are built on a model in which the users call in to the system to retrieve information. For alerting purposes, SERS needs to be able to initiate calls to the users.
Solution: Each team member can list various contact methods in SERS's "team members" database. If a team member chooses to be called when there is an alert, SERS will look up this number and place a phone call. From there, the user is prompted for a password.

Challenge: Telephony systems often employ confusing and excessive menuing. Anybody who has tried to call a vendor to get technical support for a piece of their equipment knows just how frustrating it can be to go through layer after layer of menu choices and confirmations. Also, the assignment of menu keys is inconsistent, if not plain random.
Solution: Unfortunately, to a certain extent, the menuing problem is inherent in the interaction technique and can only be minimized, not completely eliminated. To maximize the efficiency of the SERS voice menuing, detailed flowcharts were created detailing every option at every step. Then in an iterative process, all non-essential choices were "pruned" from the tree, steps were combined, and short cuts were added. Although this significantly reduced the overall size of the flowchart, it still requires eighteen, 8.5 x 11 inch pages of paper to print the basic alert notification process! Also, by flow charting the steps in detail, it was fairly straightforward to ensure functions were consistently assigned to keys (e.g., "*" to exit or "0" to return to the previous level.

Challenge: Although it is pretty simple to set up systems that allow users to select items from a menu or to enter number sequences (e.g., pass number), entering freeform text is nearly impossible from a telephone keypad. For example, using the keypad, it would be unreasonable to have a user enter general comments about an anomaly.
Solution: SERS provides the functionality to capture speech input. If a user wants to leave a "voice note" (e.g., comments, rationale), he/she presses the "#" key. SERS responds to these functions by digitizing the voice input and saving it as a ".wav" audio attachment on the appropriate Web form. Other team members can hear these notes by clicking on the ".wav" icon on the Web form.

Internet Phone/Convergence Device

"Convergence" devices are now becoming available. A convergence device is a single device (hardware and software) that performs multiple functions (e.g., a digital phone that can also provide wireless access to the Internet), thus reducing the number of devices a person needs to own.

A recent research effort under the Advanced SERS Program has been to investigate the use of such devices and to find a suitable device that could be used for paging, telephony, and a wireless Web browser access to SERS. In the fall of 1999, the first viable device was released in the US commercial market, the Qualcomm pdQ™ phone (Figure 14). The pdQ is a digital PCS cell phone handset combined with an embedded 3COM Palm III™ PDA, and the hardware and software required for integrated Web access. The Palm works exactly the same as a standard Palm (using a stylus) but includes two new applications:

1. *PDQ Alert* – a 1-way paging function that can be used to receive alerts.
2. *PDQ Browser* – a "micro browser" that provides "seamless" access to the Internet via a wireless connection.

To test the feasibility of using such a device, SERS was enhanced by creating *PocketSERS*. PocketSERS provides alert notification, response and limited Web browsing to SERS via the pdQ device or wireless-enabled PDAs.

Figure 14. Qualcomm pdQ Internet Phone

Challenge: Alert messages are limited to only a small number of characters by the wireless carriers (e.g., Sprint PCS truncates at 100 characters). This severely limits the usefulness of the data that can be sent in the alert notification.
Solution: The solution to this problem was to not send details of the alert inside the message, but rather to embed a URL inside the message. Clicking on the URL launches the Web browser and takes the user directly to the appropriate report. Figure 15 shows an alert message with the embedded URL and Figure 16 shows a PocketSERS' version of an event report.

Challenge: The screen size for the Internet phone is relatively small (a little smaller than a standard Palm Pilot) and the Web browsers rarely support useful Web features, such as frames, graphics, and some embedded tables. This makes formatting Web pages very challenging. Typically, a standard Web page will lose almost all of its formatting on such a device.
Solution: The SERS design team created multiple Web interfaces to SERS; one for PC-based browsers, and one for an Internet phone with Palm-sized displays. Unfortunately, SERS currently does not support most Internet phones because of their extremely small displays (3-5 lines of text, with 8 to 16 characters per line). Hopefully in the near future, widespread acceptance of wireless Internet standards (e.g., WAP) will make translated Web pages to unique devices more transparent.

Challenge: Accessing wireless data via a convergence device means high costs and poor coverage
Solution: For now, the resolution to this problem is out of the hands of the SERS team. A pdQ phone currently costs $800 US and 20 hours of anytime (data or voice) access costs $180 per month for Sprint wireless Web service. Also, in the United States, the lack of a standard transmission protocol (e.g., TDMA, CDMA, GSM) means that coverage for any particular phone is limited to a

specific service provider's infrastructure and agreements with other providers. Both of these obstacles should gradually disappear as the telecommunications companies expand their networks (coverage and speed) and demand and competition for data services increases.

Figure 15. PocketSERS Alert Page

Figure 16. PocketSERS Event Report

Conclusions

The new era of smaller, better, cheaper missions introduces many challenges to spacecraft operations as a greater burden is placed on the role of advanced automation. Lights-out operations is a reality, and tools that support automation of operations are critical in reducing the heavy workload imposed on reduced-staff operations. However, the ultimate effectiveness of such systems is in large part dictated by their usability. Regardless of the level of automation, at some point humans are still in the loop. The importance of creating tools with good USIs becomes even greater due to the increased challenge of keeping the part-time or multi-mission operators "connected" and "informed" as they manage more information, often from distributed locations through a variety of communications devices. The Spacecraft Emergency Response System (SERS) is one such automation tool that has faced and overcome many USI challenges. By implementing a user-centered design approach the design team optimized the USIs for these devises, resulting in a highly usable and effective system.

References

[1]Shneiderman, B. (1986) *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley.

[2] Fox, J. A., Donkers, A., Moe, K., Murphy, E., Pfister, R., Truszkowski, W., and Uehling, D. (1999). User-Centered Design of Spacecraft Ground Data Systems at NASA-Goddard. *2nd International Symposium on Spacecraft Ground Control and Data Systems (SCD II)*, Foz do Iguaccu, Brazil.

[3]Gilmore, W. E., Gertman, D. L., and Blackman, H. S. (1989) *User-Computer Interface in Process Control: A Human Factors Engineering Handbook*. Boston: Academic Press.

[4] Trimble, S. NASA Defends Failures. *Federal Computer Week*. Volume 14, No. 7. March 20, 2000.

[5]Fox, J. A., Bane, R., Baker, P., Breed, J., and Baitinger, M. (1997). Human Factors Techniques for Designing the Virtual Mission Operations Center. *The 7th International Conference on Human-Computer Interaction*, San Francisco, CA.

[6]Fox, J. A., Bane, R., Baker, P., Breed, J., and Baitinger, M. (1997). Human Factors Techniques for Designing the Virtual Mission Operations Center. *The 7th International Conference on Human-Computer Interaction*, San Francisco, CA.

[7]Fox, J. A., Starr, C., Chu, K., Baker, P., Breed, J., and Baitinger, M. (1999). Web-based Automated Reporting: Saving Time, Money and Trees. *2nd International Symposium on Spacecraft Ground Control and Data Systems (SCD II)*, Foz do Iguaccu, Brazil.